

CONTROL OF POPULATION DIVERSITY AND ADAPTATION IN DIFFERENTIAL EVOLUTION ALGORITHMS

Daniela Zaharie

Department of Computer Science, Faculty of Mathematics, West University of Timișoara
bv. V. Pârvan, no. 4, 1900 Timișoara, ROMANIA
tel./fax.: +40-256-194002, e-mail:dzaharie@info.uvt.ro

Abstract. *The desired equilibrium between exploration and exploitation in evolutionary algorithms can be achieved by controlling the population diversity. Another important issue in the design of evolutionary algorithms is the choice of parameters. Starting from these two aspects we propose a parameter adaptation for differential evolution based on the idea of controlling the population diversity. A multi-population approach of the proposed adaptive DE algorithm is also analyzed.*

Keywords: differential evolution, premature convergence, population diversity, parameter adaptation, multi-population models

1. Introduction. In all evolutionary algorithms the search for the optimum is based on two processes: *exploration* of the search space and the *exploitation* of the knowledge gathered during the search process. A good behavior of the algorithm is obtained when an equilibrium is assured between the two processes. When the exploitation process is dominant with respect to the exploration, the population loses soon its diversity and the algorithm remains into a suboptimal solution, situation called *premature convergence*. On the other hand if the exploration is dominant, the algorithm wastes much time exploring uninteresting regions of the search space. In this case a *slow convergence* could appear. Due to the specific of the recombination operator, the differential evolution algorithms (DE) introduced in [8] are very likely to prematurely converge unless their parameters are carefully chosen.

The question which arises is to *control* the relationship between exploration and exploitation such that a good convergence behavior is obtained. Ideally would be to know how to choose the evolutionary operators and their parameters to assure an *equilibrium* between exploration and exploitation. Since the exploration power of the algorithm depends on the *population diversity*, it can be viewed as a factor which influences the exploration/exploitation relationship. Thus finding the exploration/exploitation trade-off could be done by controlling the population diversity. There exists different ways of influencing the population diversity in evolutionary algorithms. Some of them are based on: (i) Injecting new information into the population (by replacing some "bad" elements with new elements, randomly distributed in the search space) when its diversity level is too low [7]. (ii) Applying alternatively mutation or recombination with selection, depending on the current population diversity [9]. The motivation is that mutation usually increases the population diversity while recombination and selection decrease it. (iii) Dividing the population into sub-populations on which independent algorithms are executed, the communication between the sub-populations being assured by a migration process. The migration can induce a "reviving" of a sub-population with low diversity [2].

On the other hand an essential role in influencing the convergence behavior of a DE algorithm is played by its strategy parameters (e.g. the population size, the magnitude of the perturbation term and the probability of applying the recombination). As many experimental results indicate ([3], [5], [8]), the performance of DE is very sensitive to the choice of strategy parameters. The process of finding by manual tuning appropriate values for the parameters is a time-consuming task, thus automating it is one of the goals in evolutionary algorithms design. Recently, in DE field have been proposed techniques for controlling the strategy parameters. In [5] a fuzzy parameter adaptation strategy is proposed. The membership functions and the fuzzy rules used in parameters modifications are established based on human knowledge and expertise gathered from previous experiments on benchmark test functions. A self-adaptive approach to DE is presented in [1] in the context of multi-objective optimization problems.

Since the parameters influence the evolution of population diversity we propose a method of adapting the parameters of DE guided by the population diversity evolution. Thus the problem of controlling the population diversity and that of parameter adaptation are combined with the final aim of inducing a good behavior of the algorithm. The adaptation method is based on some theoretical results (see [10], [11]) on the evolution of the population variance (viewed as a measure of the population diversity).

The paper is organized as follows. In Section 2 the population diversity measures are introduced and the theoretical results concerning the evolution of the population variance are reviewed. The idea of controlling the population diversity and the adaptation method are presented in Section 3. The multi-population approach and its use for the adaptive DE are analyzed in Section 4 while Section 5 presents experimental results on some benchmark test functions. Section 6 concludes the paper.

```

Initialize  $P(0) = \{x_1(0), \dots, x_m(0)\}$ ,  $g = 0$ 
Repeat
  Recombination step:
     $z_l^i = \begin{cases} \lambda x_*^i(g) + (1 - \lambda)x_{\alpha^l}^i(g) + F \cdot (x_{\beta^l}^i(g) - x_{\gamma^l}^i(g)) & \text{with probability } p \\ x_l^i(g) & \text{with probability } 1 - p \end{cases} \quad l = \overline{1, m}, \quad i = \overline{1, n}$ 
  Selection step:
    If  $f(z_l) < f(x_l(g))$  then  $x_l(g+1) = z_l$  else  $x_l(g+1) = x_l(g) \quad l = \overline{1, m}$ 
     $g = g + 1$  (increment the generation counter)
Until a stopping criterion is satisfied.

```

Figure 1: The general structure of the DE algorithm

2. Population diversity evolution in DE algorithms. We consider the problem of finding the minimum, $x^* \in D$, of an objective function $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ on which we do not impose any restriction. The DE algorithm which we shall analyze is a general one (see Figure 1) which incorporates as particular cases the DE1 and DE2 schemes introduced in [8]. The algorithm evolves a fixed size population, $P = \{x_1, \dots, x_m\}$, which is randomly initialized with elements from D . After population initialization an iterative process is started and at each iteration (generation) a new population is produced until a stopping condition is satisfied. At each generation, each population element (x_l) could be replaced (with probability p) with a new generated element. The new element is a linear combination between the best element of the population ($x_*(g)$), a randomly selected element (x_{α^l}) and a difference between other two randomly selected elements (x_{β^l} and x_{γ^l}). For each l , the indices α^l , β^l and γ^l are randomly selected without replacement from $\{1, \dots, m\}$. Besides the population size (m), the parameters of the algorithm are: $\lambda \in [0, 1]$ (the coefficient of the convex combination between the best element of the population and a randomly selected element); $p \in [0, 1]$ (the probability of replacing an element with the new generated one) and F (the factor which amplify the "differential" term).

Some frequently used particular variants of the general algorithm are [8]: DE1 (for $\lambda = 0$) and DE2 (for $\lambda \in (0, 1]$). The original DE2 scheme is characterized by the fact that $\alpha^l = l$;

To establish the relationship between the population diversity evolution and the strategy parameters we have to choose a measure for the diversity. Different measures can be used: computed either on the search space or on the objective function values space. Due to its good theoretical properties we choose as diversity measure the variance computed at the component level of the population elements. Since in DE algorithm the transformations are made independently for each component all the computations can be made at components level. To do this we will consider a population of scalars $x = \{x_1, \dots, x_m\}$ with $x_l \in \mathbb{R}$, $l = \overline{1, m}$ and we shall measure its diversity by $Var(x) = \overline{x^2} - \overline{x}^2$ with $\overline{x} = \sum_{l=1}^m x_l / m$.

To analyze the population variance evolution we introduce the following notations: x denotes the current population and Z denotes the population obtained by applying the variation operators. Since random elements are involved in the perturbation step, Z is a population of random variables. Thus we can compute only the expected value of the variance either theoretical (the theoretical mean of a random variable is denoted by $E(\cdot)$) or empirical (the empirical mean of a random variable is denoted by $\langle \cdot \rangle$). The empirical expected value of the population variance is computed by averaging the variance over all components.

In [11] a relationship between the parameters and the variance modification after the perturbation step has been obtained:

Theorem 1 *The expected variance of population obtained after recombination is:*

$$E(Var(Z)) = \left[2pF^2 + \frac{(1-p)^2}{m} + \frac{m-1}{m}(p(1-\lambda)^2 + (1-p)) \right] Var(x) + \frac{m-1}{m} p \lambda^2 (1-p) (\overline{x} - x_*)^2. \quad (1)$$

The proof of this result is based on classical results from probability theory concerning the mean and the variance and can be found in [11].

When $\lambda = 0$, the best element of the population is not taken into consideration and (1) becomes

$$E(Var(Z)) = \left(2pF^2 + 1 - \frac{2p}{m} + \frac{p^2}{m} \right) Var(x) \quad (2)$$

which is the result presented in [10]. When $p = 1$, all the population elements are replaced with the new

generated elements and (1) becomes:

$$E(\text{Var}(Z)) = \left(2F^2 + \frac{m-1}{m}(1-\lambda)^2\right) \text{Var}(x). \quad (3)$$

In both these particular cases a simple linear dependence exists between $E(\text{Var}(Z))$ and $\text{Var}(x)$. This remark suggests the idea of the adaptation method which we propose.

3. The controlling idea and the adaptive algorithm. The controlling method is based on the result on population variance evolution presented in the previous section. Let us consider that $x(g)$ is the population obtained at generation $g-1$ (in the same time it is the initial population for generation g). During the generation g it is transformed in $z(g)$ (by recombination) then in $w(g)$ (by the selection process). $w(g)$ will represent the starting population for the next generation, $x(g+1)$. When $\lambda = 0$ or $p = 1$ relations (2) and (3) imply that:

$$E(\text{Var}(Z(g))) = c(m, p(g), F(g), \lambda(g)) \text{Var}(x(g)).$$

On the other hand, during the algorithm execution we can compute r_g which satisfies: $\text{Var}(x(g+1)) = r_g \text{Var}(x(g))$ (while the population did not lose all its diversity, $r_g = \text{Var}(x(g+1))/\text{Var}(x(g))$). The computed value, r_g , gives us information about the variance tendency: if $r_g < 1$ the variance is decreasing and if $r_g > 1$ the variance is increasing. The controlling idea is to choose the parameters F , p or λ such that the recombination applied in generation g compensates the effect of the previous application of recombination and selection. This means to choose $F(g+1)$, $p(g+1)$ or $\lambda(g+1)$ such that

$$c_{g+1} \stackrel{\text{not}}{=} c(m, p(g+1), F(g+1), \lambda(g+1)) = \gamma \frac{\text{Var}(x(g))}{\text{Var}(x(g+1))} \quad (4)$$

where $\gamma > 0$ is a *new* control parameter of the variance evolution. Thus after applying the recombination in generation $g+1$ (using the new parameters) we expect that $\text{Var}(z(g+1)) = \gamma \text{Var}(x(g))$. The adjusted parameters will have a "repairing" effect on the algorithm behavior in the following sense: (i) if population variance is decreasing ($\text{Var}(x(g+1)) < \text{Var}(x(g))$) then $c_{g+1} > \gamma$, thus the next recombination can compensate the decrease of variance; (ii) If population variance is increasing ($\text{Var}(x(g+1)) > \text{Var}(x(g))$) then $c_{g+1} < \gamma$ thus the next recombination can compensate the increase of variance.

By choosing an adequate value for γ we can impose a desired behavior. If $\gamma = 1$ then $E(\text{Var}(Z(g+1))) = \text{Var}(x(g))$ which means that the recombination in generation $g+1$ compensates the effect on population variance of operators applied in generation g . Choosing $\gamma > 1$ one can compensate a high decrease of the variance, thus we can avoid premature convergence situations. Choosing $\gamma < 1$ one can compensate an increase of the variance, thus we could accelerate the convergence but with the risk of inducing premature convergence.

The idea of the parameter adaptation is to solve, with respect to F and p or λ respectively, one of the following *critical equations*:

$$1 - \frac{2p}{m} + \frac{p^2}{m} + 2pF^2 = \gamma \frac{\text{Var}(x(g))}{\text{Var}(x(g+1))}, \quad (5)$$

$$\frac{m-1}{m}(1-\lambda)^2 + 2F^2 = \gamma \frac{\text{Var}(x(g))}{\text{Var}(x(g+1))}. \quad (6)$$

Equation (5) can be solved with respect to F and p and it is used when $\lambda = 0$. Equation (6) can be solved with respect to F and λ and it is used when $p = 1$ (or very near to 1). Since from a single equation we can find only a single unknown, the parameters will be adapted asynchronously: at each generation only one parameter will be determined the other being considered fixed. To simplify the notations we will denote the right hand side of (5) and (6) with c . By solving (5) with respect to F one obtains the following adaptation rule for F :

$$F = \begin{cases} \sqrt{\frac{m(c-1)+p(2-p)}{2mp}} & \text{if } m(c-1) + p(2-p) \geq 0 \\ F_{\text{inf}} & \text{if } m(c-1) + p(2-p) < 0 \end{cases} \quad (7)$$

with F_{inf} the minimal value for F . A sufficient condition for increasing the population variance by recombination is that $F \geq 1/\sqrt{m}$, thus we shall use $F_{\text{inf}} = 1/\sqrt{m}$. An upper bound for F can also be imposed (empirical results suggest $F_{\text{sup}} = 2$). By solving (5) with respect to p one obtains the following adaptation rule for p :

$$p = \begin{cases} -(mF^2 - 1) + \sqrt{(mF^2 - 1)^2 - m(1-c)} & \text{if } c \geq 1 \\ p_{\text{inf}} & \text{if } c < 1 \end{cases} \quad (8)$$

with p_{inf} the minimal value for p . Since $p \in [0, 1]$ a minimal value for p should be near 0, e.g. $p_{\text{inf}} = 0.01$ while the maximal value should be $p_{\text{sup}} = 1$. Solving eq. (6) with respect to F and λ one obtains similar adaptation rules for the case when $p \simeq 1$. When the computed values of the parameters are outside their domain they are

corrected as follows: a value less than the lower bound is replaced with the lower value while a value greater than the upper bound is replaced with the upper bound.

Since the variance is computed at component level, there exists n relations (5) and (6) which allow the adaptation of the sets of parameters $\{F_i\}_{i=\overline{1,n}}$, $\{p_i\}_{i=\overline{1,n}}$ and $\{\lambda_i\}_{i=\overline{1,n}}$. The algorithm in Figure 1 will be modified as follows. The parameters F , p and λ are replaced with F_i , p_i and λ_i , at each generation are computed the variances for all n components and, after the selection step, the parameters are adapted (using variances computed during the current and the previous generations). If we want to adapt two parameters (for instance F and p or F and λ) at each generation only one of them is adapted (for instance at odd generations are computed the values $\{F_i\}_{i=\overline{1,n}}$ while at even generations are computed the values $\{p_i\}_{i=\overline{1,n}}$). Unlike other adaptation rules (e.g. self-adaptation) where each individual has specific values of the parameters the proposed method uses different values for each component. In this way the particularities of the fitness landscape could be "captured" by the parameters adaptation process.

4. The influence of migration on population variance. Another method to keep the population diversity at a acceptable level, mainly used in coarse grained parallelization of evolutionary algorithms, is to work with different sub-populations. On each sub-population a DE algorithm is executed for a given number of generations. To assure the communication between sub-populations a migration process takes place: some elements from each sub-population are replaced with elements from others sub-populations. Thus a sub-population with a low diversity can be "revived" after the migration takes place. For genetic algorithms, different migration strategies have been proposed and theoretically studied [2]. The existing parallelization of DE are based either on evaluating in parallel the objective functions for the population elements [4] or on the cellular model [6].

For the DE algorithm we propose to divide the population in s sub-populations of the same size, μ . The communication is assured by the following migration strategy: each element from each sub-population can be switched (with a given *migration probability*, p_m) with a randomly selected element from a randomly selected sub-population (including the sub-population which contains the initial element). Thus the migration topology is based on random connections. To analyze the influence of the migration on the sub-population variance we shall introduce the following notations. The notations and the analysis is still at component level. Let $(x_1^q, x_2^q, \dots, x_\mu^q)$ be the sub-population q ($q = \overline{1, s}$) before migration and $(V_1^q, V_2^q, \dots, V_\mu^q)$ the sub-population after migration (due to the random character of the migration, V_i^q are random variables). Let $\overline{x^q}$ and $\text{Var}(x^q)$ be the mean and the variance of the sub-population q , respectively. Also let \overline{x} and $\text{Var}(x)$ be the mean and the variance of the entire population, respectively. By using classical properties of the mean value and of the variance we obtained that the expected variance of sub-population V^q satisfies:

$$(1 - p_m)^2 \text{Var}(x^q) + p_m^2 \frac{\mu - 1}{\mu} \text{Var}(x) \leq E(\text{Var}(V^q)) + 2p_m(1 - p_m) \frac{\mu - 1}{\mu} \overline{x^q \overline{x}} \leq (1 - p_m) \text{Var}(x^q) + p_m \frac{\mu - 1}{\mu} \text{Var}(x). \quad (9)$$

Relation (9) clearly suggests that when a sub-population lost its diversity ($\text{Var}(x^q) = 0$), if the entire population is still varied ($\text{Var}(x) > 0$) by choosing a high enough migration probability ($p_m \simeq 1$) one obtains $E(\text{Var}(V^q)) > 0$. Since the adaptive algorithm modifies the parameters according with the variance evolution, the implementation of a multi-population adaptive DE does not need specific choice of the DE parameters (besides γ , which has a rather intuitive meaning). Numerical results (see Table 3) suggests that a serial multi-population implementation of the adaptive DE can overcome some difficulties which arise for some test functions (e.g. Rastrigin).

5. Experimental analysis. In the experimental analysis we tried to find answers to the following questions:

- Is the adaptive DE effective and what is the influence of the new control parameter, γ , on its behavior ?
- Can improve the multi-population approach the behavior of the adaptive DE ?

To answer the first question we developed a statistical analysis of the adaptive algorithm based on the test functions described in Table 1. During the experiments, the following parameters were fixed: $m = 50$ (the population size), $n = 30$ (the problem dimension), $r = 50$ (the number of independent runs of the algorithm, used to compute the averaged values). The single parameters F , p and λ from the classical DE were replaced by sets of parameters $\{F_i\}_{i=\overline{1,n}}$, $\{p_i\}_{i=\overline{1,n}}$ and $\{\lambda_i\}_{i=\overline{1,n}}$, each component having its own parameters. The initial values of the parameters are randomly chosen in their domain. At each generation a set of parameters is adjusted by using the corresponding adaptive rules (e.g. F_i are modified as in (7)). The stopping condition which we used is: " $f_* < \epsilon$ or $\langle \text{Var}(x) \rangle < 10^{-12}$ or *the number of generations is greater than a maximal value*" (f_* is the best value of the objective function found into the population). The algorithm is considered successful (*Success* in Tables (2),(3)) and if the first part of the condition is satisfied and it prematurely converges (*PC* in Table (2),(3)) if only the second part is true. If the third part of the condition is true then it is considered that a slow convergence situation occurs (*SC* in Table (2),(3)). The maximal number of generations was set to 5000. In Table (2) for each situation is presented the number of cases (out of 50) and the corresponding averaged

number of generations. Results in Table (2) suggests that small values of γ can accelerate the convergence with the risk of inducing a premature convergence while high values of γ induces slow convergence. For the test functions f_A and f_G similar results have been obtained with the variant based on adaptation of F and λ ($p \simeq 1$), while for f_R it doesn't work very well.

Figure 2(a) illustrates the evolution of a parameter F (it corresponds to the first component) when $p = 0.5$ is fixed and the initial value was $F_0 = 1$. It is clear that greater values of γ leads to greater values of the parameter. Figure 2(b) illustrates the simultaneous evolution of F and p starting from some not very good initial values: $F_0 = 2$, $p_0 = 0.1$. We can remark that the adaptive rules bring very quickly (in the first generations) the parameters to adequate values. Figures 3(a) and (b) illustrate the evolution of the averaged population variance and of the best objective function value, respectively. It is easy to see that small values of γ lead to a fast decrease of the variance and in the same time of the function values while high values of γ lead to a slow decrease of the variance.

In the multi-population implementation we divided the population into s sub-populations. On each sub-population an adaptive DE algorithm (with its own set of adjustable parameters) is executed. The implementation was on a serial machine and after each 100 generations a migration takes place. The results in Table (3) suggest that we can avoid premature convergence without increasing γ obtaining in this way a better convergence rate.

Name	Expression	Domain
Rastrigin	$f_R(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
Ackley	$f_A(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$
Griewank	$f_G(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^n$

Table 1: Test functions

γ	Ackley				Griewank				Rastrigin			
	Success nr/gen	PC nr/gen	SC nr/gen	$\langle f_* \rangle$	Success nr/gen	PC nr/gen	SC nr/gen	$\langle f_* \rangle$	Success nr/gen	PC nr/gen	SC nr/gen	$\langle f_* \rangle$
0.5	29/601	21/671	—	0.04	39/495	11/947	—	10^{-2}	12/691	38/837	—	0.97
0.75	42/652	8/701	—	10^{-4}	50/530	—	—	10^{-5}	17/689	33/856	—	0.91
1	50/2260	—	—	10^{-5}	50/1911	—	—	10^{-5}	50/2018	—	—	10^{-5}
1.25	—	—	50/5000	10^{-2}	—	—	50/5000	10^{-2}	—	—	50/5000	0.27

Table 2: Convergence behavior of the adaptive DE ($\lambda = 0$, $\epsilon = 10^{-5}$, p and F are adaptive and randomly initialized.)

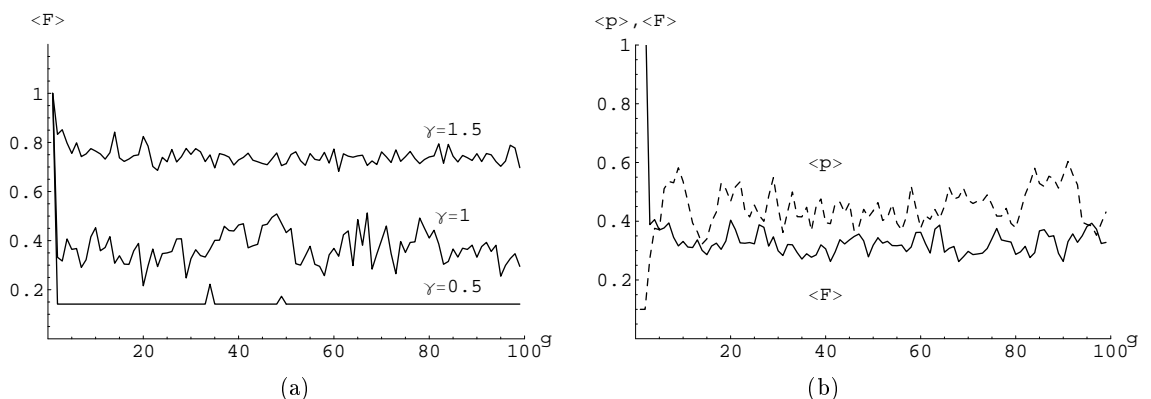


Figure 2: Parameters evolution ($\lambda = 0$, $m = 50$); (a) adaptation of F for different values of γ ($p = 0.5$, $F_0 = 1$); (b) adaptation of p and F ($p_0 = 0.1$, $F_0 = 2$).

6. Conclusions. The numerical results imply that a adaptive DE is more effective than the self-adaptive variant based on the idea used in [1] (we obtained comparable results only for the sphere model). The parameter γ allows us to control the decreasing rate of the population variance. The numerical experiments suggest that there exist a relationship between the decreasing rate of the population variance and the convergence rate. Thus by an adequate choice of γ we can control the convergence rate. It seems that we replaced the problem

of choosing the initial parameters with the problem of the choice of γ . It is true, but instead of choosing two interrelated parameters we have to establish only one.

The multi-population approach offers us a new way of avoiding the premature convergence thus it can be used for the adaptive DE mainly when we want to use small values of γ . Besides this gain offered by the serial implementation of the multi-population adaptive DE this approach opens the possibility of an easy parallel implementation.

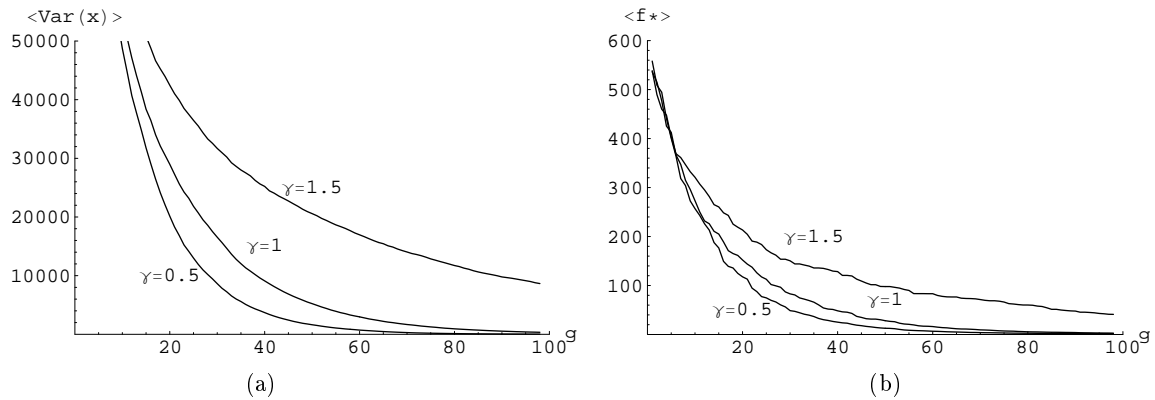


Figure 3: Influence of γ on the algorithm behavior ($\lambda = 0$, $m = 50$, $p = 0.5$, F adaptive); (a) averaged population variance at component level; (b) averaged best value of the fitness function

γ	s	μ	p_m	Success	PC	$\langle f_* \rangle$
0.9	1	60	0	21/824	29/992	0.7959
0.9	2	30	0.5	42/982	8/1125	0.1989
0.9	3	20	0.5	49/1117	1/1208	0.0199
0.9	4	15	0.5	48/1215	2/1670	0.0398
0.9	5	12	0.5	50/1302	—	0.000009
0.9	6	10	0.5	50/1311	—	0.000009

Table 3: Influence of migration on the adaptive DE. Test function: Rastrigin, $\epsilon = 10^{-5}$, $\lambda = 0$, F and p adaptive

References

- [1] H.A. Abbass; The Self-Adaptive Pareto Differential Evolution Algorithm, *Proceedings of 2002 Congress on Evolutionary Computation*, pp. 831-836, 2002.
- [2] E. Cantú-Paz; Topologies, migration rates, and multi-population parallel genetic algorithms, In Banzhaf, W. et al. (Eds.) *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, CA: Morgan Kaufmann, pp. 91-98, 1999.
- [3] R. Gämperle, S.D. Müller, P. Koumoutsakos; A Parameter Study for Differential Evolution, in A. Grmela, N.E. Mastorakis (eds.) *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, WSEAS Press, pp. 293-298, 2002.
- [4] Lampinen J.; Differential Evolution-New Naturally Parallel Approach for Engineering Design Optimization, in Barry H.V. Topping(ed.), *Developments in Computational Mechanics with High Performance Computing*, Civil-Comp Press, Edinburgh, pp.217-228, 1999.
- [5] J. Liu, J. Lampinen; Adaptive Parameter Control of Differential Evolution, in R. Matoušek, P. Ošmera (eds.), *Proc. of Mendel 2002, 8th International Conference on Soft Computing*, pp. 19-26, 2002.
- [6] M. Salomon; Parallélisation de l'évolution différentielle pour le recalage rigide d'images médicales volumiques, RenPar'2000, Besançon, 2000.
- [7] T. Šmuc; Improving Convergence Properties of the Differential Evolution Algorithm, in R. Matoušek, P. Ošmera (eds.), *Proc. of Mendel 2002, 8th International Conference on Soft Computing*, pp. 80-86, 2002.
- [8] R. Storn, K. Price; Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Technical Report TR-95-012*, ICSI, 1995.
- [9] R.K. Ursem; Diversity-Guided Evolutionary Algorithms, *Proc. of Parallel Problem Solving from Nature VII (PPSN-2002)*, 2002.
- [10] D. Zaharie; Critical Values for the Control Parameters of Differential Evolution Algorithms, in R. Matoušek, P. Ošmera (eds.), *Proc. of Mendel 2002, 8th International Conference on Soft Computing*, pp. 62-67, 2002.
- [11] D. Zaharie; Parameter Adaptation in Differential Evolution by Controlling the Population Diversity, in D. Petcu et al. (eds), *Proc. of 4th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania, pp. 385-397, 2002.